

**REMARKS**

In the Office Action<sup>1</sup>, the Examiner rejected claims 1, 3-18, and 20-37 under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent 6,169,984 to Bogdan et al. ("Bogdan") in view of U.S. Patent 6,366,920 to Hoose et al. ("Hoose").<sup>2</sup>

By this reply, Applicants amend claims 1, 3-5, 9-18, 20, 22, 23, and 26-31; cancel claims 6-8; and add new dependent claim 38. Claims 1, 3-5, 9-18, and 20-38 are pending in the application. Support for the claim amendments and new claim can be found, for example, at paragraphs 25, 33, and 35 of the specification.

**Interview Summary**

As an initial matter, Applicants would like to thank Examiner Augustine for the courtesies extended to Applicants' representative during the telephonic interview of October 26, 2009. During the interview, Applicants representative and the Examiner discussed proposed claim amendments in view of the outstanding prior art rejection. The Examiner agreed that amending the claims as proposed would place the application in condition for allowance.

The claim amendments and remarks herein are consistent with the proposed claim amendments and discussion during the interview.

---

<sup>1</sup> The Office Action may contain a number of statements reflecting characterizations of the related art and the claims. Regardless of whether any such statement is identified herein, Applicants decline to automatically subscribe to any statement or characterization in the Office Action.

<sup>2</sup> Although the Office Action contains no statement indicating which claims are rejected and the grounds upon which the claims are rejected, it appears that the Examiner intended to reject each of pending claims 1, 3-18, and 20-37.

**Rejection under 35 U.S.C. § 103**

Applicant respectfully traverses the rejection of the pending claims under 35 U.S.C. § 103(a) as unpatentable over Bogdan in view of Hoose.

Amended independent claim 1 recites a computer program product comprising instructions for performing a method comprising, among other steps,

grouping the displayed user interface elements . . .  
**alphabetically according to characters contained in  
respective text labels** associated with the user interface  
elements. . . [and]

when the user navigation input is the backward user  
navigation input . . .

shifting input focus in the reverse direction to a previous  
user interface element in the current group having an  
associated text label matching the key press of at least  
one text character **when the current group is the same  
as the target group, and**

shifting input focus to a user interface element last in  
order in the target group **when the current group is not  
the same as the target group**

(emphasis added). Bogdan and Hoose, taken alone or in combination, fail to disclose or suggest a method comprising at least the claimed “grouping” and “shifting” steps.

**The Claimed “Grouping” Step**

At page 4, the Office Action concedes that Bogdan fails to disclose or suggest the claimed “grouping” step. The Office Action then relies upon Hoose to remedy the deficiencies of Bogdan. Page 4.

Hoose discloses “a method and system . . . for allowing the user to enter data into fields in a non-sequential manner, in order to expedite the entry of the data.”

Abstract. In Hoose, “[o]nce a mode file is activated, the user can move the cursor to

preselected fields **in a preselected order** by pressing the tab key 19A on the keyboard.” Col. 3, ll. 56-57 (emphasis added). “A mode file contains a listing of fields and their order.” Col. 3, ll. 49-50.

In other words, Hoose uses a mode file to customize or predetermine an order in which the application cycles through input fields when the user presses the “tab” key on the keyboard. Contrary to the assertion in the Office Action, however, Hoose is completely silent with respect to “grouping the user interface elements . . .

**alphabetically according to characters contained in [their] respective text labels”** (emphasis added), as recited by amended independent claim 1. Even if the preselected tab ordering of user input fields described in Hoose’s mode file could be deemed a grouping of user interface elements, Hoose’s grouping would still not be “alphabetically according to characters contained in respective text labels . . . ,” as recited by amended independent claim 1. Instead, Hoose’s grouping would be based on the user’s (arbitrary) preselected or customized order of fields indicated in the mode file.

At page 4, the Office Action points to the discussion of Tables 1 and 2 in columns 5 and 6 of Hoose as disclosing the claimed “grouping” step. In the cited portions, however, Hoose only explains that Table 1 lists the preselected sequence of input fields defined by the mode file, that is, the sequence of input fields the application will cycle through when the user presses the “tab” key. Specifically, the “Entry Field Charac. Posn.” column of Table 1 indicates the sequence, going down through the rows. For example, the first input field in the sequence is “field-1-char-1,” the second input field in the sequence is “field-2-char-1,” the third input field in the sequence is “field-3-char-1,” etc.

Accordingly, as can be seen from Table 1, Hoose's sequence of input fields listed in the mode field is arbitrary and based on the preferences of the user. The sequence is not based whatsoever on "**characters contained in respective text labels** associated with the user interface elements," as recited by amended independent claim 1. Accordingly, Hoose does not "group[] the user interface elements . . . **alphabetically according to characters contained in [their] respective text labels**" (emphasis added), as recited by amended independent claim 1.

Moreover, Table 2 of Hoose only "provides a listing of the objects that contain the desired data entry fields and also provides information on dependence . . . between objects." Col. 6, 11-14. Table 2, however, does not disclose a "grouping [of] the displayed user interface elements . . . alphabetically according to characters contained in respective text labels associated with the user interface elements," as recited by amended independent claim 1.

### **The Claimed "Shifting" Steps**

Bogdan discloses "[a] user interface control implementing a global keyboard navigation system by way of printable keyboard character input to extend an incremental type search across all window components within a defined scope." Abstract. In Bogdan, the user interface elements are grouped hierarchically into containers. See, e.g., col. 5, ll. 27-35. Bogdan performs "a recursive search of all parent and child generations [within the defined scope] . . . to find a match between the first character of keyboard input from the user and the first character of a component name." Col. 7, ll. 27-30.

In a telephone discussion on September 29, 2009, the Examiner clarified that the Office Action's allegations with respect to the claimed "backward user navigation input." Specifically, the Examiner indicated that the Office Action alleges at pages 3 and 4 that the description in column 9 of the ALT key modifier corresponds to the claimed "backward user navigation input." Specifically, Bogdan discloses, "if it is determined . . . that the ALT key modifier was present along with the input character . . . then . . . the present scope of input is broken and **the input focus is redirected to the top level component among the set of components** . . . ." Col. 9, ll. 41-46 (emphasis added).

Even assuming the combination of the ALT key modifier and the input character constitutes the claimed "backward user navigation input," which Applicants do not concede, Bogdan discloses that this input combination automatically redirects input focus "to the **top level component** among the set of components." Col. 9, ll. 45-46. For example, in Fig. 2 of Bogdan, component 210 is the "top level component" within the scope of container 200. See col. 5, ll. 38-39.

Thus, when the user presses the ALT modifier in combination with the input character, Bogdan always automatically redirects input focus **to the highest-level component in the container**. In contrast, amended independent claim 1 provides for two different scenarios for backward navigation input. Specifically, in amended independent claim 1, input focus is shifted "to a previous user interface element in the current group . . . when the current group is the same as the target group," **and** input focus is shifted "to a user interface element last in order in the target group when the current group is not the same as the target group." Because Bogdan always redirects input focus to the top level component in the container, however, Bogdan cannot

disclose or suggest “shifting input focus . . . **to a previous user interface element in the current group . . . when the current group is the same as the target group,**” and shifting input focus “**to a user interface element last in order in the target group when the current group is not the same as the target group**” (emphasis added) as recited by amended independent claim 1.

Applicants also note that Bogdan discloses that the user can navigate through the components conventionally using the direction keys. See col. 6, ll. 40-46. Even if a user in Bodan uses the direction to navigate forward and backward through the components, such input to the direct keys does not comprise “a forward user navigation input comprising **a forward modifier key press combined with a key press of at least one text character . . .**, and a backward user navigation input comprising **a backward modifier key press combined with a key press of at least one text character . . .**” (emphasis added), as recited by amended independent claim 1.

Hoose fails to remedy the deficiencies of Bogdan with respect to the claimed “shifting” steps of claim 1. As set forth above, Hoose provides a mode file indicating predetermined sequence of user input fields through which an application cycles when the user presses the “tab” key. Hoose, however, also fails to disclose or suggest at least “when the user navigation input is the backward user navigation input,” “shifting input focus in the reverse direction to a previous user interface element in the current group having an associated text label matching the key press of at least one text character when the current group is the same as the target group,” and “shifting input focus to a user interface element last in order in the target group when the current group is not the same as the target group,” as recited by amended independent claim 1.

For at least the above reasons and, as agreed during the interview, amended independent claim 1 is allowable over Bodgen in view of Hoose. Amended independent claims 9, 13, 18, 23, and 27, while of different scope from claim 1, recite similar features to those discussed above in connection with claim 1, and are therefore allowable over Bogdan in view of Hoose for at least similar reasons as claim 1. The remaining claims are allowable due to their dependence from one of the independent claims, as well as for their own recited features. For example, amended dependent claims 14 and 28 further recite “**automatically executing an action associated with the user interface element** when it is determined that the key press of at least one text character uniquely identifies the user interface element” (emphasis added).

At page 13, the Office Action points to column 10, lines 25-33 of Bogdan as disclosing “detect[ing] a sequence of one or more characters that uniquely identifies an activation user interface element.” In the cited portion, Bogdan discusses how the user can search for a specific component that has the same prefix as another component using keyboard input. Specifically, Bogdan discloses, “the user can search for the third occurrence of a component having the prefix ADMIN by typing ADMIN followed by a second N and then a third N. Thus, the user can implement a more specific component name search . . . by retyping A numerous times to step through all occurrences of component names that start with the letter A.” Col. 10, ll. 25-33.

While Bogdan may allow the user to **search** through multiple components having the same prefix to uniquely identify a component having the same prefix as another component, Bogdan does not disclose or suggest “**automatically executing an action associated with the user interface element** when it is determined that the key press

of at least one text character uniquely identifies the user interface element” (emphasis added) as recited by amended dependent claims 14 and 28. Accordingly, dependent claims 14 and 28 are allowable over the prior art for at least this additional reason.

In addition, new dependent claim 38 further recites

receiving a display of a graphical user interface, the graphical user interface including the user interface elements;

identifying the user interface elements in the display of the graphical user interface; and

extracting from the display text labels for the identified user interface elements.

Applicants respectfully submit that at least these additional features are neither disclosed nor suggested by the prior art. Accordingly, new dependent claim 38 is allowable over the prior art for at least this additional reason aside from its dependence from independent claim 18.

### **Conclusion**

In view of the foregoing, Applicants respectfully request reconsideration of this application, the withdrawal of the rejection, and the timely allowance of the pending claims. If the Examiner believes a telephone conference would be useful in resolving any outstanding issues, the Examiner is kindly invited to contact the undersigned at 202.216.5118.



Please grant any extensions of time required to enter this response and charge  
any additional required fees to our Deposit Account 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,  
GARRETT & DUNNER, L.L.P.

Dated: October 29, 2009

By: 

James D. Stein  
Reg. No. 63,782  
202.216.5118